Ulla Maasio

Customer Data Mapping Documentation Process

Helsinki Metropolia University of Applied Sciences

Degree Bachelor of Engineering

Information Technology

Thesis

1 April 2013

Metropolia

Tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Ulla Maasio Asiakasdatakuvauksen dokumentaatioprosessi 50 sivua + 4 liitettä 1.4.2013
Tutkinto	insinööri
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	
Ohjaajat	lehtori Olli Hämäläinen palvelupäällikkö Jani Leppänen

Insinöörityön tarkoitus oli suunnitella ja dokumentoida loogisen datamallin ja asiakashallintajärjestelmän tietokannan sekä laskutusjärjestelmän tietokannan välinen kuvaus. Insinöörityöhön sisällytettiin myös prosessikuvaus ja kuvaukseen osallistuvien roolien väliset interaktiot.

Työ tehtiin tutkimalla loogista datamallinnusta ja prosesseja. Työn aikana osallistuttiin kokouksiin järjestelmien toimittajan kanssa, haastateltiin toimittajan työntekijöitä ja testattiin tietokannasta toimittajan tuottamaa aineistoa. Prosessi kuvauksen luonnille ja sen ylläpitämiselle tehtiin tutkimalla, miten työ siirtyy työntekijältä toiselle työn eri vaiheissa ja vertaamalla sitä ITIL -prosesseihin.

Insinöörityön aikana loogisen asiakasmallin ja tietokantojen kuvaus tehtiin onnistuneesti tietoliikenneyritykselle, mutta insinöörityössä se kuvataan fiktiivisen yrityksen kautta. Prosessikuvaukset auttavat selkeyttämään, miten työvaihesiirtymät hoidetaan laadukkaasti.

Asiakastietomallin ja tietokantojen kuvauksen lopputuloksesta voidaan analysoida, pitääkö asiakasmallia parantaa ja siihen liittyviä järjestelmiä kehittää liiketoimintavetoisempaan suuntaan.

Avainsanat	looginen malli, datan kuvaus, liiketoimintakriittisen perustiedon hallinta, ITIL -prosessit, dokumentaatio

Metropolia

Abstract

Author Title Number of Pages Date	Ulla Maasio Customer data mapping documentation process 50 pages + 4 appendices 1 April 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	
Instructors	Jani Leppänen, Service Manager Olli Hämäläinen, Senior Lecturer

The purpose of this thesis was to describe planning and documenting mapping which takes place between the logical data model, the databases of a customer relationship management system and of a billing system, used by businesses. The thesis also covers the workflow, related processes and interactions between parties participating in the mapping.

The final year project was carried out by studying logical data modeling and processes. The final work also included participating in meetings with the system supplier, interviewing the employees of the vendor and testing and verifying data provided by the vendor from system databases. The process of mapping creation and mapping maintenance was studied by observing how work was transferred from one person to another in the different phases of the mapping work. This was compered to ITIL processes.

During the final project, customer data model and database mapping was done successfully for a telecommunication corporation, but in the thesis they are described through a fictive corporation.

The process flow of the mapping clarifies how different work phases are performed with high quality. Due to analyzing the result of the mapping, the logical customer data model can be improved and the related systems can be developed towards a business driven direction.

Keywords	Logical model, data mapping, master data, ITIL processes, documentation

Contents

Abbreviations and Terms

CRM	Customer Relationship Management
BSS	Business Support Systems; product, customer, order management system
DB	Database is an organized collection of data
DBMS	Database Management System; software to create, maintain and retrieve DB data
DCL	Data Control Language to authorize users to access and manipulate data
DDL	Data Definition Language managing data structure
DML	Data Manipulation Language to modify data
DQL	Data Query Language for retrieving and transferring data
HR	Human relations
ISO	International organization for standardization
IT	Information Technology
ITIL	Information Technology Infrastructure Library collection of guidance processes for IT
KPI	Key Performance Indicator
LDM	Logical Data Model
MDM	Master Data Management, business critical data management process
MOF	Microsoft operations Framework is practical guidance for IT
MoM	Moments of Meeting aka minutes of the meeting
OSS	Operational Support Systems used interacting with network system
PDM	Physical Data Model
RDBMS	Same as DBMS but specially used in Relational Databases
ROI	Return of Investment

SQL	Structured Query Language, language specifically created to manage Relational Databases
TOGAF	Open Group Architecture Framework methodology for defining information systems
UI	User Interface
XML	Extensible Markup Language is a markup language encoding documents in human and machine readable format

1 Introduction

Companies are storing customer information in their customer relationship management systems (CRM), and for creating invoices of the ordered services and products they are using billing systems. Finding relations between business understanding of the customer and how this information is located in the system is called data mapping.

The scope of this work includes mapping between the logical model, data model of the CRM system and data model of the billing system. CRM and billing applications exist independently and have their own data.

The purpose of this document is to show how mapping documentation can be done and how to create a process flow for maintaining this documentation. This is achieved by providing basic information about data, data handling, and information about documentation and documentation systems which can be used to keep data mapping up. Processes and roles participating in creating and maintaining the mapping will be introduced as well. At the beginning of this thesis, basic concepts are introduced and at the end a process flow that can be used to create such a mapping is presented, together with a practical example of process flow usage.

As a work assignment I had a chance to participate in a data model mapping project. Static data which is used for different purposes and within different processes across the corporate organization is called master data. Part of the organization is taking care of data convergence and reliability. This master data management (MDM) group of company where I work had completed earlier a "Customer Data Model design" project which gave the baseline for the mapping. The Customer Data Model design focused on what data is business critical and how MDM could be applied to it so that this data forms a solid and homogenous platform to be used by different parts of the organization. This mapping project was a more technical continuation of the data model design project which was business-oriented. Information gathered in this work project will be used here to describe an imaginary company data mapping project to

fulfill the assignment of creating mapping documentation and a process for keeping up this documentation.

This document can be of assistance to people working with business data on a daily basis as it covers the basic concepts related to data and its mapping: what mapping contains, which processes are related and how changing a data model reflects on mapping and vice versa.

This information is also useful to employees who need to use, create or maintain data mapping between a data model and databases. The target groups are data architects, designers, developers, and people working with master data management (the purpose of MDM is, for example, to take care of static data of the customer and products), who already have a good IT knowledge, but need some refreshing of the importance of documenting and of business processes. It also helps people who are using data for different purposes, including testers, data analysts, and teams on the ongoing service operations side such as service desk by providing basic information on the structure of the system. This thesis also introduces some basic ideas about databases. In addition it gives ideas on how data mapping can be used in ongoing service operations work as some people working with it might not have IT training and since most of the IT related work is outsourced.

2 Master Data Management

In general, application data structure is defined based on the functionality of the application. Business organizations have two kinds of usage of data: executing transactions by business processes and supporting operational activities. There are software applications designed to support operational processes within each area of the business, with their own data definitions, data dictionaries, table structures, and application functionality. Thus, every area of the business typically has its own applications, databases and needs to use data.

> *Master data tend to exist in more than one business area within the organization, so the same customer may show up in the sales system as well as the billing system. Master data objects are used within transactions, but the underlying object sets tend to be static in comparison to transaction systems and do not change as frequently.*
> [2,8.]

One example of master data is customer information. The same customer information can be held in different systems within the company, for different uses. Examples of the uses might be interacting with customers by phone, selling products or sending ordered products. The same data can be used when sending an invoice to a customer, for analysing customer interests for company products, for developing new products or for marketing purposes.

How can data be used more efficiently? How is it ensured that the same information about the customer is kept synchronized throughout different systems? Achieving this requires a change from system focused thinking into centralized thinking of managing data. At first it is necessary to identify different meanings of data and how they can be integrated into a view which is available across the organization for different purposes. Of course, this is not easy to manage. If there are several systems with databases, consolidation of systems is needed. Consolidating systems are beneficial from a financial point of view and asset managing point of view and of course consolidation supports green values with reduced hardware and less electricity.

Master data management aims to combine people, processes and technology so that they work together to get data to meet the organization's requirements. This means appointing roles and responsibilities to the people who are working with data, appointing ownership of the data and instructions on how to handle data and develop

systems and tools to take constantly care of the data. There are some tools for MDM that help to improve data quality and define Key Performance Indicators (KPI) for monitoring data quality. These tools will be introduced later in chapter 4 on handling data.

3 Customer Data Model

One role of MDM is to define a standardized data structure across the organization. The data model is a bridge between common world understanding of data and its form in IT systems. To support business needs, the creation of the data model is started by defining the meaning of information to the business and its importance.

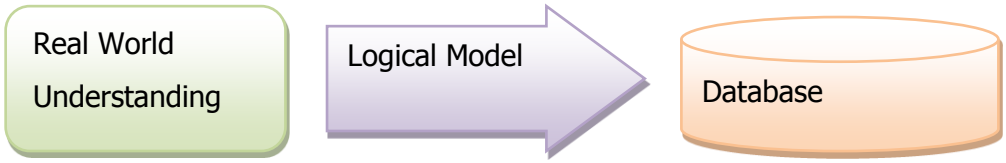


Figure 1. Data model location between real world understanding and database

In figure 1 it can be seen that in the beginning of the process the business critical objects must be identified and described based on business terminology. This forms the conceptual design. The translation layer between conceptual design and stored data is the data model. In a logical data model the business objects are described as entities and entity attributes. After entities and attributes are defined, they can be directly translated to database tables and columns in ideal circumstances. When dealing with existing databases, these entities must be identified in the existing database objects.

Understanding the dataflow and business rules, which are valid across various systems, makes it easier to handle master data for different purposes. The same data can be used for keeping up customer information and communication with the customer, for collecting information and analyzing it for marketing purposes as well as for developing new services for the customer. Understanding of the business process and of the data model is important for keeping the master data as harmonized as possible. This can prevent financial losses and allows keeping high a quality of the services and interaction with the customers, generating more trust on their side.

3.1 Conceptual Design

Resolving business needs to have a data standard starts with understanding the business requirements. With this knowledge, a common terminology and the identified

critical high-level business entities and relations between them are created. Conceptual design, also known as high-level data model, is used to communicate core data concepts, rules and definitions to be part of overall application management [6,48]. The conceptual model is independent of any physical data storage considerations.

There are some core questions the business world is trying to answer when studying who exactly is the customer and all the relations with the customer, how customer is defined and what is important about the customer.

- Who? Who are we potentially in a relation with (these are called parties or stakeholders)?
- Where? Where is this party physically located?
- Who are our customers?
- How? How do we communicate with our customers?
- What kind of hierarchy and roles do customers have? Where in this hierarchy does a party belong?

The answers lead to a common high level understanding of the customers and entities the organization interacts with and their core elements.

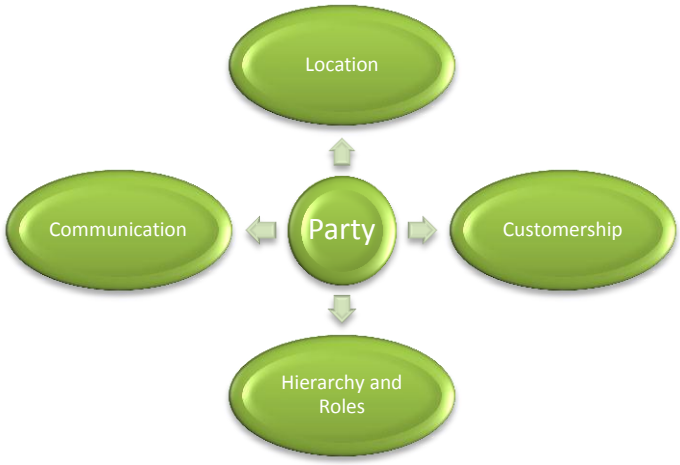


Figure 2. Conceptual design of customer model

Figure 2 presents a sample of conceptual design or a customer model. This can be used as a common starting point for business and IT teams when creating and presenting the high level data model. Here it can be seen that a party is the central entity and has relations to other entities judged as most critical from the point of view of the business processes.

The party is the core of the customer model. An agreement with the party might not exist, but the party might have been contacted or information on it might have been collected from a third party. The party entity can be considered to be a real person, organization, branch, subsidiary, legal entity or holding company.

The party has several other elements around it. It might have an address which answers the question about where the party is located. This information is needed when sending letters, invoices and devices to a customer. The party might have several addresses for different purposes: several offices, shipping addresses and one for invoicing if the headquarters of the company are located in a different place.

The customer entity describes who our customers are, the parties with whom the business has an active agreement. For example, in a family, the father could have made an agreement for a service used by his son. In this scenario, the father is a customer and a party while the son is just a party. This information can also be used for marketing purposes as well as for rewarding customers (loyalty campaigns).

The "hierarchy and roles" entity describes the relations various parties have with customer entities. A party might be a business which has different departments or locations. In this case it is necessary to define business legal structure as well as the geographical structure.

The communication entity tells how and when communication takes places with customers and parties. This contains information about preferred ways of interacting with the customer.

3.2 Logical Design

The next step is to translate the conceptual data model into a logical data model (LDM) and in this phase the database requirements are taken into account. This is a business area specific data model which identifies business critical relations and attributes.

> *The LDM is a graphical representation of the people, places and things of interest to the organization (entities), the core business rules governing them, the relationships between them and all of the characteristics about each of them that are needed to run the applications of the business. The LDM has no dependences on physical implementation. [6,53.]*

LDM has more details compared to a conceptual data model. The conceptual data model such as the logical model is constructed based on business processes but it presents more detailed information. Attributes are defined for each entity but no particular implementation details are stored for them (such as constraints or data types).

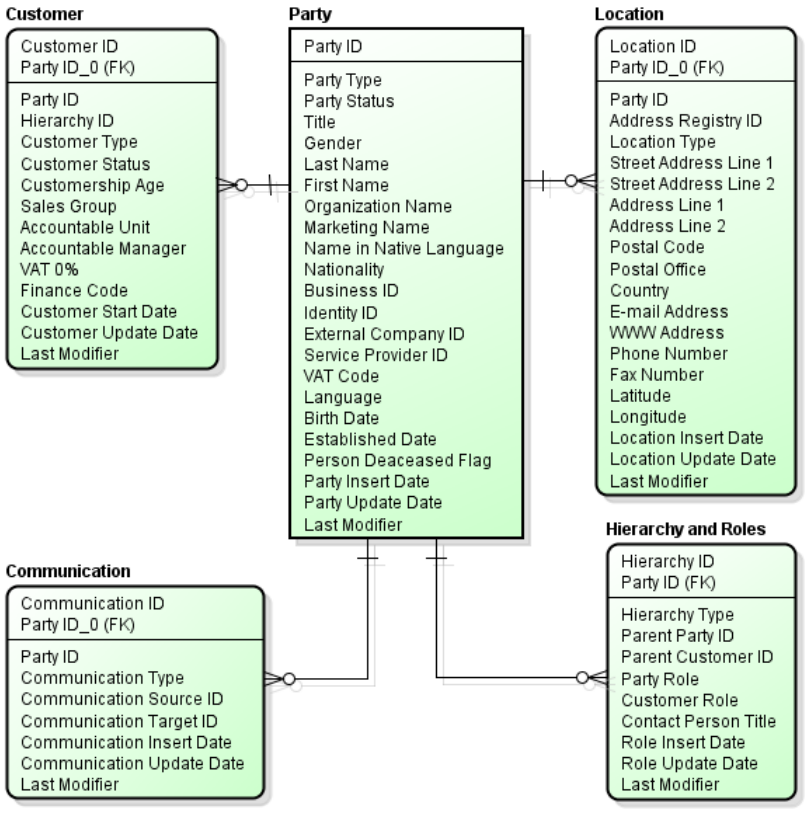


Figure 3. Logical design of the customer model

The LDM in figure 3 contains the conceptual model entities with some sample named attributes. It is not mandatory to include definitions of allowed attribute values in the

LDM. "One to many" relationships are presented as a fork line at the entity which has a relationships several times with the other entity.

3.3 Physical Design

A physical data model (PDM) is a graphical representation of the physical structure of an actual database or data structure. The logical data model of the customer is translated into software data storage terms to match the needs of the applications using the data and databases storing it. When there is a fresh start, the logical model is the starting point based on what is chosen for a database type and what kind of tables and columns will be created into databases.

Storage media is created as well as database specific relations, indexes and security guidelines [7,175]. Database architects design database configuration and schemas based on the above considerations.

4 Storing Data

Large amounts of data are handled in everyday life. Data can be an audio file which is stored on a CD or contact details which are written in an address book or a name of a book in a library search engine.

To be able to manage this amount of data, the data needs to be stored in computer systems. Data is separated, at the lowest level being stored as bits and bytes in computer memory. The smallest independent and meaningful unit of stored data is called a data element. Logically related data elements are grouped into records. Collections of records are called files. A repository for data in electrical format is called a database (DB) and a database management system (DBMS) maintains data files.

There are several things to be considered when storing data in a database, for example, how it is presented on a logical level and how it will be stored in the media, what kind of media is selected and what kind of operating and database system it will have and how and when backup of the database is done. The physical data structure is built and each table is mapped to a physical structure on a disk drive where it will be stored.

There are different kinds of models of databases such as hierarchical databases (tree structure), network databases (tree structure which allows multiple parent and child records), object databases (information is presented in the form of object and is used in object-oriented programming) and relational databases. The relational database is the base solution.

Also, there are different types of databases for different purposes such as:

- Active database which is used for security monitoring, alerting and authorization
- Operational database which stores business operational data with high volumes of updating transactions at all times

Data Warehouse stores data from operational databases and third party systems. It differs from the operational database in the sense that it fetches data from source systems rarely. For example an incremental update is done once a day and a full update is done once a week. Data Warehouse is used mainly for different kind of

reporting and data analyzing. Relational databases are most commonly used and popular vendors are IBM, Microsoft and Oracle, which is also offering the open source system MySQL.

4.1 Database Management System

A Database Management System (DBMS) is software used to create, maintain and retrieve data from a database. DBMSs are built to work with one of the database models presented above.

A DBMS provides the following facilities:

- Abstraction, data can be presented in different ways (views), independent of the actual storage on the media
- Independence, providing a layer between the data and applications using it, preventing changes of the logical and physical structure of data
- Access control, enabling and managing access to use stored data files for users and applications
- Security management, preventing unauthorized users from viewing or updating the database or parts of it
- Data integrity enforcement, providing rules that govern the type of data that can be stored in fields and columns
- Concurrency control management, to be able to handle multiple users and applications accessing and modifying records. Locking mechanisms are used to prevent data inconsistency
- Ensuring atomicity, all-or-nothing quality in transactions in case of system crash in the middle of transaction
- Ensuring recoverability, in case of database failures it has capability to recover and restore from backup files
- Database logging, capturing information about data modification

The benefit of DBMS is its ability to maintain and query large amounts of data while assuring data integrity and consistency.

4.2 Relational Database

A relational database is built on the relational model. This model groups data fields in tables and defines constraints on the possible values and combinations of values. The relational model was invented in 1970 by E.F.Codd [11,22] and it is based on relational algebra.

Table 1. Example of the relational database table

ID (Integer)	FirstName (String)	LastName(String)	SSID(String)
1	Ulla	Maasio	123456-124U
2	Cristian	Doe	231434-233J

Relational database tables are sets of tuples that have the same attributes (data type, constraints). In table 1 there is an example table which represents one relationship between various data elements. Attributes have a relation to each other and are set in a row. In the Contact table, a tuple is formed of an attributes's ID, FirstName, LastName and SSID. Attributes have values of data types such as integer, string, and date. A primary key uniquely identifies a tuple within the database. In table 1 ID is set as a primary key and it identifies uniquely any of the two sets of attributes. Relational DBs are used with relational database management system (RDBMS) software.

4.3 Database Schema

A database schema presents the organization of data and relationships within the database and it separates the physical and logical aspect of data representation.

Database schemas are divided into different levels:

- Internal schema presents physical storage describing how and where the data are organized in physical data storage.
- Global or conceptual schema presents the stored data structure defined by database model.
- External schema presents views created for different groups of users.[12.]

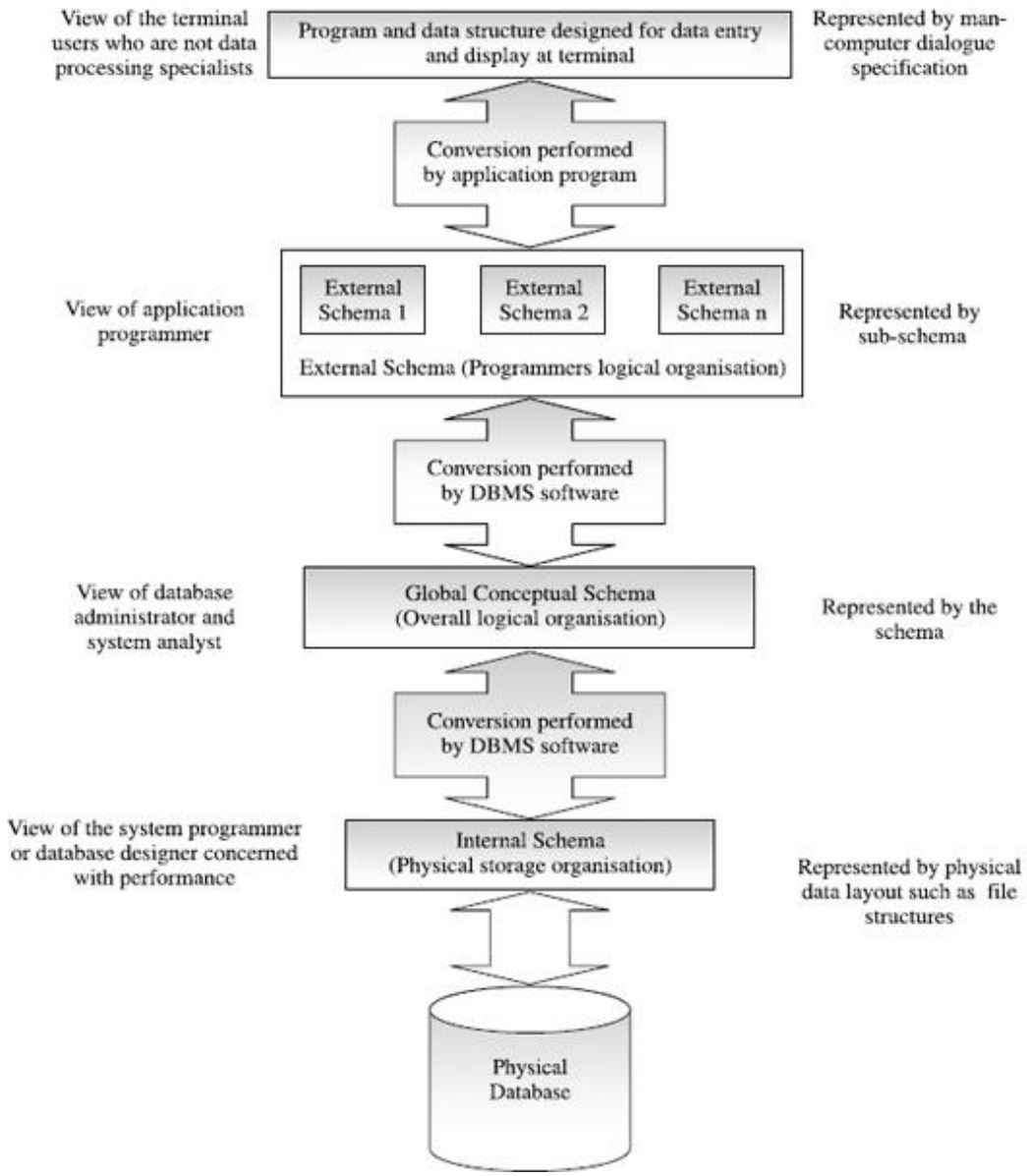


Figure 4. Database organization [12, Fig.1.15]

Figure 4 shows how users have different kinds of access levels and how data is seen and used by different user groups.

4.4 SQL – Structured Query Language

To be able to manage relational database data, Structured Query Language was created. It has its roots in the SEQUEL language which was developed by Donald D. Chamberlin and Raymond F. Boyce in early 1970's. [14,9.]

SQL has different "dialects" depending on the database and vendor which it is used for, but the basic elements are the same, based on the SQL:2003 standard. Standards have been enhanced since 2003 and lots of different standards are valid for SQL. SQL was designed for data storage, retrieval and manipulation. [11,26-27] It can be divided into four parts for managing data:

- Data Manipulation Language (DML) modifying data
- Data Definition Language (DDL) managing data structure (such as tables and index)
- Data Query Language (DQL) retrieving and transferring data
- Data Control Language (DCL) authorizing users to access and manipulate data (SQL standard do not include DCL)

The DML is to add, update and delete data in a table.

```
INSERT INTO Contact
            (FirstName,LastName,SSID)
    Values('Ulla','Maasio','123456-124U');
```

The DDL is to create, alter, rename, drop and truncate table.

```
CREATE TABLE Contact(
    ID   INT,
    FirstName   String(40),
    LastName   String(40),
    SSID   String(11),
    PRIMARY KEY (ID));
```

The most used statement is the Select query, used to retrieve data from the database based on a set of search filters with DQL. Data can be retrieved from one table, multiple tables, filtered according to rules and results that can be ordered.

```
SELECT * FROM Contact
WHERE FirstName LIKE 'Cris%';
```

DCL is designed for providing or denying access to data to different parties (users) using the database.

```
GRANT SELECT,UPDATE
      ON Contact
          TO User_Anna;
```

DCL is not standardized, but most database vendors have it in their product.

5 Handling Data

Isolated data fields do not provide much value, but a set of data can be defined as information and using this information can be used for improving knowledge. To be able to provide information from stored data, this data must meet the expectations of data quality set by the business needing it. Data mapping is one of the tools that can be used for data examination and improvement and it works together with other tools. The data mapping is needed for performing the data improvement activities and the mapping serves as input to issues found while creating data mapping. Some of these tools, which are used by the MDM group, will be introduced later in this chapter after presenting the overall dataflow.

5.1 Dataflow

In this chapter the dataflow of generic business is examined using a communications operator (such as mobile phone) as an example. The dataflow of the business has several different layers that can be grouped in Business Support Systems (BSS) and Operational Support Systems (OSS).

The main areas of BSS are product management, order management, customer management and revenue management.

OSS supports processes such as maintaining network inventory, provisioning services, configuring network components, and managing faults.

When a customer makes an order through a self-service channel, such as the operator's web page, the order is delivered to the CRM system and from there to the Billing system (BSS) and after that it is activated or provided by the operator’s infrastructure or network via the OSS. Processing a customer’s invoices is done in the billing system within BSS.

Applications have their own checks and rules for user data input and changes will not be allowed if conditions are not met. On the other hand, every application has their own logic and might convert data to meet the application's internal requirements so that the original data form is lost and information might change in the dataflow.

Sometimes there are system failures that might lead to data corruption or data not being propagated to all systems.

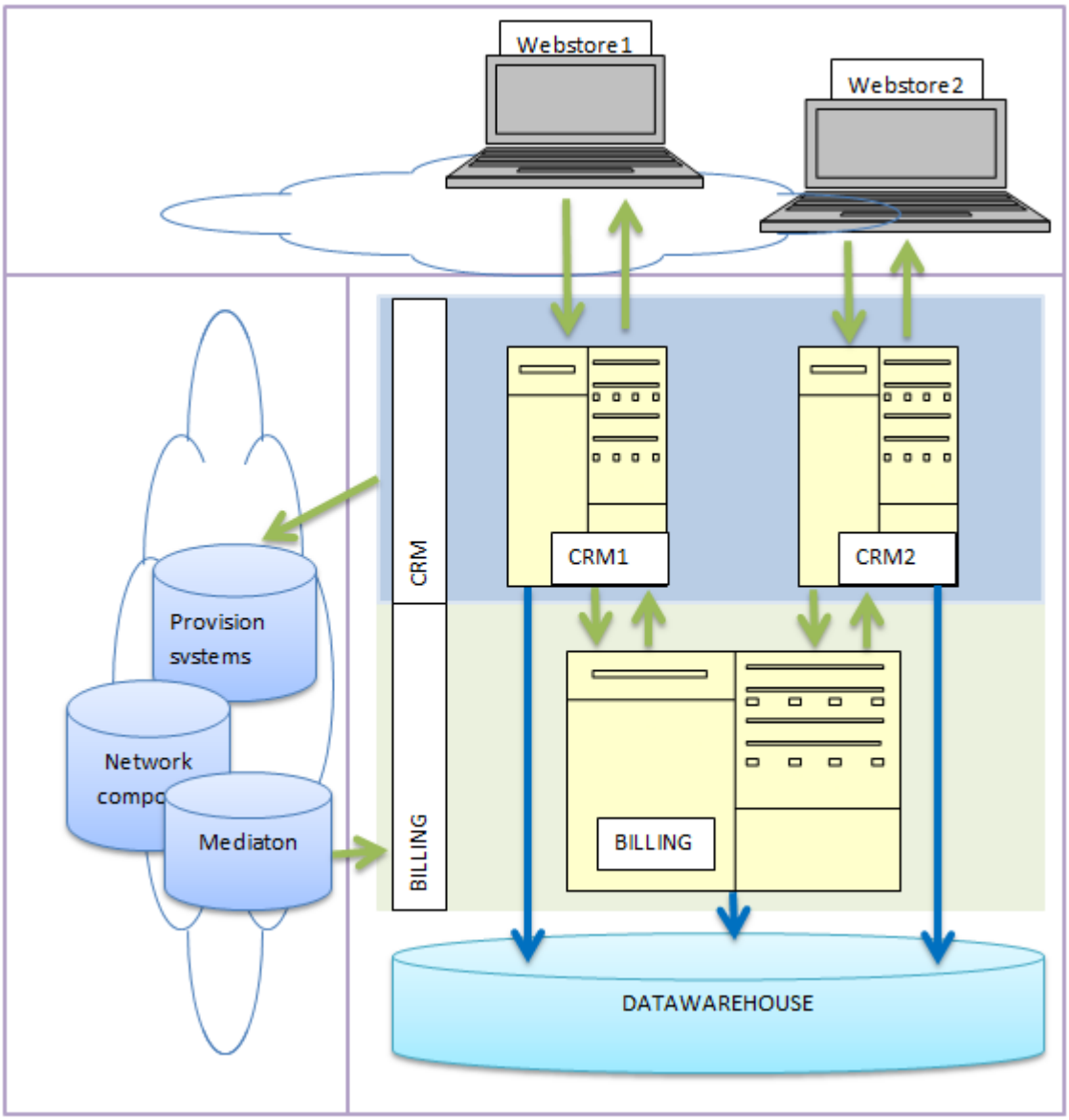


Figure 5. Customer dataflow from end user system to OSS and BSS and back

Applications have their internal data delivery processes (see figure 5, CRM-Billing). Data can also be injected by external systems using APIs (see figure 5, WEB2 store – CRM) or sent to other systems in a common format (such as XML). Also other systems might get data from systems to use data for different purposes.

5.2 Data Profiling

Profiling data refers to one time analysis of the data to assess its state in terms of cleanliness, patterns and missing fields [3].
Usually data profiling is done before data is migrated from one system to another (for example when replacing one system with another), trying to determine risks and the current state of system data for further handling before migration.

5.3 Data Parsing

Parsing identifies recognizable patterns. It enables the automatic recognition and subsequent standardization of meaningful value components, for example, the way of showing and storing a phone number as +358501234567 instead of 050 123 4567 or 050 1234567. [4,1.]

5.4 Data Standardization

Parsing results in defined patterns, expressions and grammars to make rules for distinguishing between valid and invalid data values in the database [4,31]. Data values must be stored in particular formats to be usable by all the systems.

Also there is a need for common data terminology to allow accurate and efficient communication between different teams and systems. Different business units may use the same terms with different meanings, and the IT systems have their own terminology.

5.5 Data Cleansing

Data cleansing refers to general data harmonization and maintaining data quality. It is built on parsing, standardization and enhancement tools. This means eliminating obsolete information, parsing values of known error patterns and merging duplicates [4,32].

Legal regulations restrict the information which a business can hold about its customers. If a private customer has no longer an active agreement with the business after two years, then the business should delete this customer's information from its database.

5.6 Data Enhancement

Data enhancement is a data improvement process that adds information based on third party systems[4,33]. Typically this can be done by completing addresses with the postal code from external systems like Itella, in Finland.

5.7 Identity Resolution

Identity resolution is the automated merging process which selects the best data values from records which are duplicated and selects the "best copy" of the data which can support operational processes and analytics to establish a single view of the customer [4,32].

6 Documentation

This chapter explains various aspects of IT documentation as it is a relevant part of this thesis (data mapping documentation). Technical documentation is an important part of engineering as it holds important information and allows the distribution of this information to others. Unfortunately many documents are written in vain as they contain information which is aimed to the writer only. At the same time security issues have to be taken into account, because documents may contain sensitive information.

There are different kinds of technical documents. The most common ones are:

- Technical requirements which contain statements that identify attributes, capabilities, characteristics or qualities of a system
- Architecture or design which includes relations to an environment and construction principles to be used in design of software components
- Manuals for the end-user, system administrators and support staff

A good document is written for a specific target group and it also includes information which can be used beyond the target group. Documents are seldom read in detail so they should include a meaningful index and key words so that one who is seeking information can easily find it.

There are some common guidelines for documentation which are presented below.
Documents can exist in various formats, such as written documents on paper, as media audio files stored electrically in a document library or, the most common format, as text files to be viewed on a computer.

6.1 Documentation Processes

The high level documentation processes include creating a document, updating a document and reviewing a document.

In an organization there are different groups of people which need to have access to a document in order to be able to review it or make changes to it. To be able to keep track of document changes and to avoid cases in which changes overwrite other

changes, only one person should be able to modify a document at a given time. For that, the lock functionality is usually used.

Different organizations and individuals have their own practices of storing documents. Some use shared folders and manually updated document versions, others use dedicated documentation management systems. Document handling should have the same policy throughout the organization to ensure uniform document maintenance and to allow the automation of related tasks. There are different kinds of document managing systems that allow people to access documents based on security guidelines.

It is a good practice to review documents from time to time and keep the information current.

6.2 Version Control

The purpose of version control is usually historical tracking of changes which were done on the documented matter. It also gives the possibility to go back on document history and restart updating the document from there. Version control is also called reversion control.

Version control systems can be divided into centralized and distribution based systems. A centralized version control system has one centralized server managing files and users do changes in the server. In a distribution based version control system everyone has their own copy of a repository that is cloned to user hard drive and changes are pulled from server or pushed to it so they can be modified or shared with others.

A document goes through four different stages of reversion control:

- Baselining
- Changing
- Identifying differences between versions
- Storing into repository

The reviewed and approved document version is called baseline. In this stage the document should also be signed and dated. Changes are clearly identified, each

version is differently marked to distinguish it from the others and notes are added describing the changes done since the previous version.

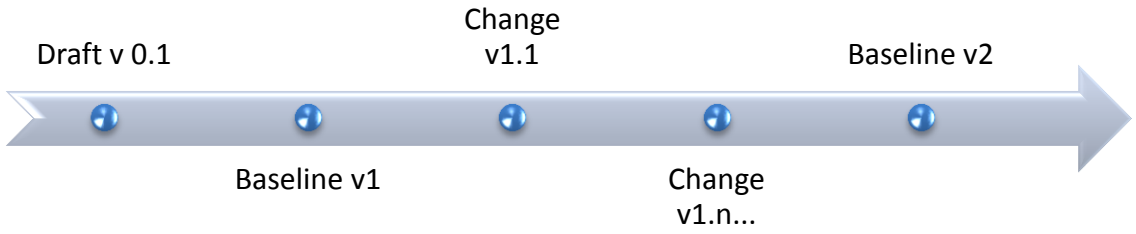


Figure 6. Versioning process of the document

When the document is reviewed (see figure 6), it is marked as doc.v1. This becomes the baseline of the document. If changes are made, subversions doc.v1.n and so on are created and finally, when the changes are approved, a new version is created, doc.v2.0. Documents need to have a storage place, which is called a document repository.

6.3 Security Guidelines

Organizations usually define their own security levels on documentation to mark which groups have reading and writing access to documents. When creating a document, the security guidelines are considered as well and the document's security policy is marked.

There are four main types of security levels for documents:

- Public documents are open documents for everyone to see
- Internal documents are intended for company employees only
- Confidential documents contain sensitive information and are targeted to a specific group
- Secret documents contain business critical information and are usually available to only top management

Documents need to be stored in such a way that the target group can easily access them and parties outside the group are blocked. Access to the document is granted based on approval from the document owner.

Backup of documents should be done on a regular basis. Because of that, it is not advisable to keep documents on personal workstations. A common secure repository should be used where access levels are defined beforehand and backups of the system are taken on regularly. Also care should be taken when sharing a document. For example sending a document by e-mail should be carefully considered as that exposes it to unwanted recipients, either because the receiver will send it forward to other parties or because the email communication outside the organization can be intercepted. It is good to remember that e-mail is considered as public communication.

6.4 Document Management Systems

Some employees prefer storing documents on their own computers and some departments have shared disk space for storing documents. Storing documents on one's own computer is rarely a good idea since it increases the risk of document loss, risk of not having the most updated version and risk of improper document sharing.

A document management system (DMS) is used to track and store electronic documents and images of paper documents. It is also capable of keeping track of the different versions modified by different users.

Microsoft SharePoint 2010 has a document library where documents can be shared and it has capabilities such as versioning, document approval and integration.
It has the following features:

- Check-in/check-out, only one user at the time modifying the document
- Versioning, history tracking
- Approval (revision)
- Workflows
- Information management polices (security policies)
  [8,674.]

Other vendors offer DMS as well, such as Open Source Kordil with similar functionality.

Another option instead of DMS to document information is the wiki technique. A Wiki is a website which can be modified by users via a web browser using markup language. Most commonly known public wiki is www.wikipedia.org. For corporation use there are private wikis as well.

Wiki pages can be structured with hypertext links to other pages. Their advantage is that they are a collaboration tool where authorized users can edit them from different locations and the changes are instantly visible.

A wiki's change log includes reversion history and differences between previous page versions can be studied.

A wiki needs a wiki system to be run. The system includes a web browser, web servers, a file system and RDBMs. The most commonly used wiki software application is MediaWiki which is free open source software [15,8]. There is also commercial wiki software such as Confluence.

Storing documents in a proper way is usually neglected. One reason for it is lack of information about documents and procedures since different user groups' documents are stored differently and with different policies. Also, many times documents are stored by a user on computers for ease of access. A DMS has advantages, as all users have the most updated information and distribution is easy.

Some documents are created and maintained by different development groups and their existence and location are not known to other groups that need the information. A good practice is to have a master document of interfaces and references to other documents. The referenced resources should be easily accessible according to user permissions.

7 Process

A process is a series of steps that need to be executed in a particular sequence to achieve a goal. Observing actions and writing them down helps with understanding a chain of events and through that processes can be improved. Like any other complex activity that involves several parties, creating and maintaining a data mapping should follow a well-defined process.

There are different tools for improving processes. These are also predefined processes which can be used as a reference for the process being executed:

- The Open Group Architecture Framework (TOGAF) describes a methodology for defining information systems in terms of a set of building blocks. It is used commonly on applications, data and technical architecture.
- Microsoft Operations Framework (MOF) is practical guidance for everyday IT practices and activities, helping users establish and implement IT services.
- ISO/IEC 20000 is a standard setting out the requirements for an IT Service Management System.
- The Information Technology Infrastructure Library (ITIL) is a set of practices for IT service management that focuses on aligning IT services with the needs of business. It describes procedures, tasks and checklists that are not organization-specific.

All of these methodologies have their advantages and are designed to support IT operations. Since ITIL has the advantage of having both business perspective and IT perspective and it is widely spread, it is the one discussed further and the base for our data mapping processes. From business perspective, having updated data mapping is an IT service that provides a view of the relations between business entities and software systems.

7.1 The Information Technology Infrastructure Library

ITIL's core guidance is divided into six publications: Introduction to ITIL Service Management Practices, Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Management.

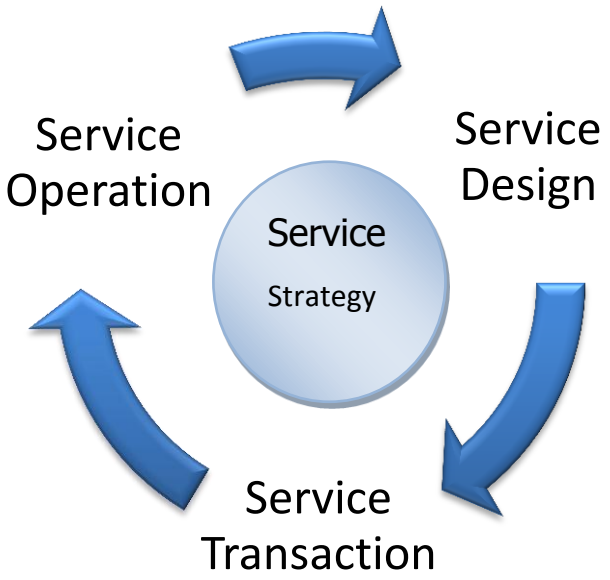


Figure 7. ITIL Framework basic processes are a continual flow

Each of the basic elements of an IT service is part of the flow of an ongoing process as it is described in figure 7. The service is created based on a business need, it is being deployed and, it is improved to be more efficient based on monitoring results.

7.2 Basic Processes of ITIL

The base of ITIL is Service Strategy which contains guidance on how to view service management from the point of view of organizational capability and strategic assets.

Service Strategy leads to models for Service Design, Service Transaction (service implementation) and the service operation. All phases are improved by the Continual Service Management based on feedback.

The service design meets the need for management guidelines for creating a service and for converting requirements into portfolios (of services and service assets). At the time of service design the impact on the overall service, the management systems and tools (which include the architecture and the technology) are considered.

The service transaction process guidelines explain the releasing of the designed service into production. It builds upon current data, assets and media and provides a management model which is based on the service strategy.

Day to day management processes are described in the service operation process. It describes how to proactively and reactively monitor the service. By measuring data incidents, incidents can be proactively prevented, and if something happens the problem management process can be used to correct it. Strategic objectives are realized through service operation.

Quality and change management and capacity improvement are described in continual service improvement. Data monitoring is done in continual service improvement, and based on it, it can be decided if data de-duplication, data cleansing or enrichment need to be done.

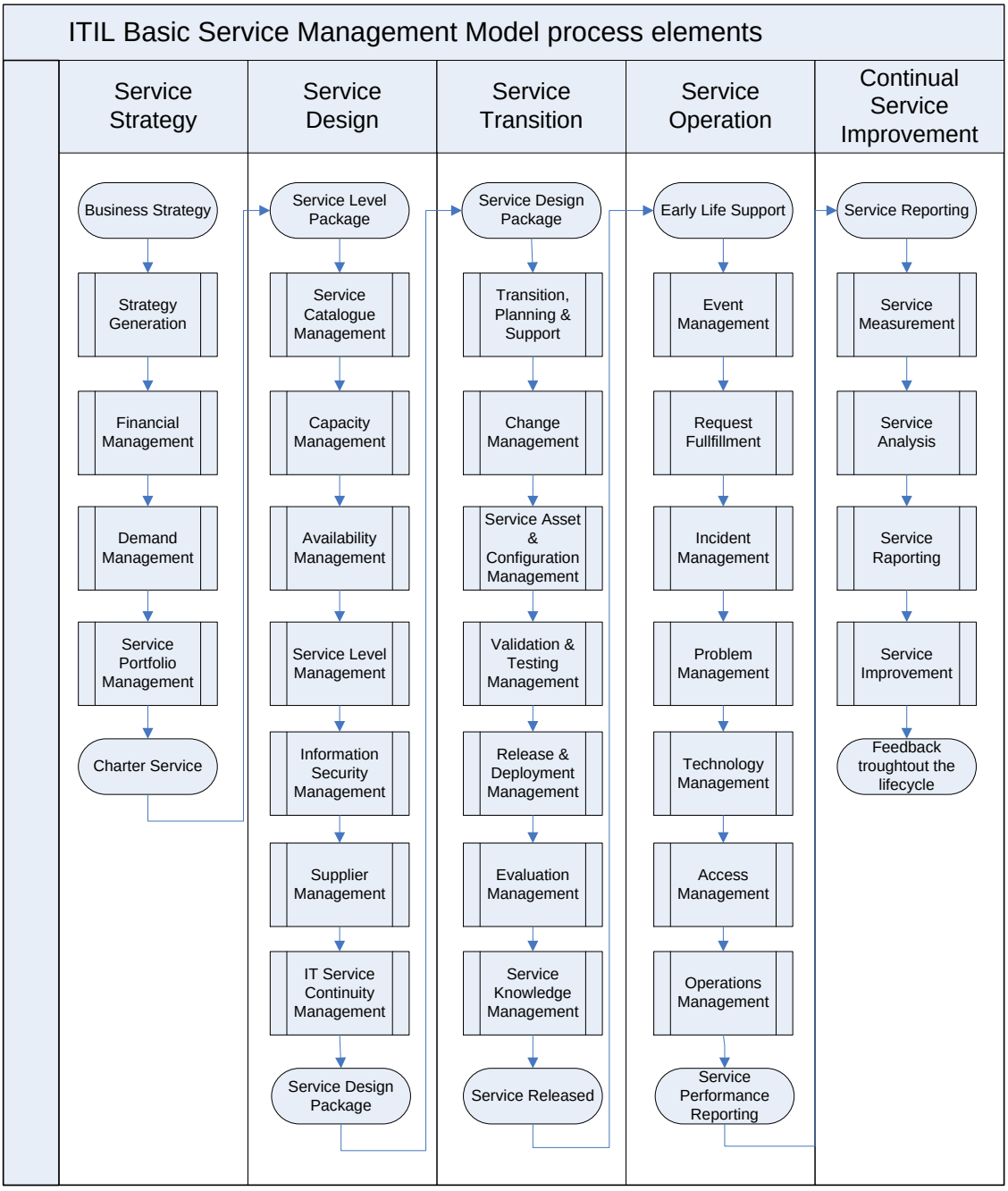


Figure 8. ITIL Basic Service Management process elements [10:154]

Each of the main process flows has dependencies on some of the others and they form a continuous flow as shown in figure 8.

8 Process to Document Customer Data Mapping

Building up and maintaining an application requires several documents. The main benefit of the documentation is that it creates common understanding of the application and its functionality as well as of a purpose. Documents are written by the people who develop or work with the application on a regular basis and have an overall view of it.

Documents are stored usually in a DMS, but some teams prefer creating wiki-pages and build up a document repository with wiki-page structure. Wiki-page structure has its advantages but a DMS can store more efficiently scanned paper documents, word documents or PDF files which people are accustomed to.

8.1 Data Mapping

Data mapping begins with the need to know where a data element is located in two distinct data models. Data mapping is usually understood as the mapping between an application and the database. This is because, traditionally, when an application is being developed, entities and attributes in the application are mirrored by tables and columns that store application data. Instead of mapping from application to database, MDM maps LDM to the database. This eliminates the complexity of the dataflow between the systems and just a high level understanding of this flow is needed. Based on this idea, different kinds of views can be created from the database to be used by the business teams.

Data mapping is done by comparing the entity attribute information from both the conceptual and the logical model with the corresponding element or elements in the database. To help understand the mapping process it is better to use a top-down thinking method while analyzing the application dataflow. This means that a logical model element attribute is chosen and then the corresponding field on the application side is identified. When the application field is found, it is much easier to discuss the logical attribute with the application vendor or developer for locating the attribute on the database level.

In the dataflow chapter it was presented that data flow from the self-service channel to CRM, then to the Billing system and from both applications to the Data Warehouse. In its simplest form, data mapping can be done with spreadsheets but there are also specialized tools which do data mapping from LDM to database or vice versa.

When the mapping information spreadsheet is created, it needs to include data on two levels, logical level and physical level, as mapping is done between them. At the physical level, if there are several systems holding data, columns need to be added for each. A third level, user interface(UI) is optional but useful for an easier understanding since most of the parties using the mapping are familiar with the UIs of the applications.

At the logical level:

- Logical model Entity's Attribute name
- Logical model Entity's Primary key
- Logical model Entity's Attribute type (if defined)
- Logical model Entity's Attribute length (if defined)
- Logical model Entity's Attribute description

At the physical level:

- System in which the information can be found
- Table in the database
- Field of the table
- Type of the field
- Length of the field
- Format
- Description from the application's point of view
- Mandatory / non-mandatory from the application's point of view
- SQL Query on the database that will return the attribute data
- An example of attribute data

At the UI level:

- UI Field
- Picture of the UI Field
- Notes

A spreadsheet is efficient for documenting data. As more information is needed to explain system behavior and references to other documents, it is good to consider creating a standard text document.

8.2 Benefits of Customer Data Mapping Documentation

The customer data model mapping presents how business requirements are met by the actual data stored in databases. Replication of data, data structure or functionality mismatches can be noted and corrected.

Customer data mapping enables consolidating information by integrating data contained in different systems in single views.

Documentation also gives a better understanding of the way business entities are implemented in the IT systems. This allows business side teams to consider system limitations and system features not used at their full potential.

Customer data mapping documentation can be used during testing, incident solving and system development when verifying data is needed. It also serves as a starting point for dealing with the data stored in the systems for employees that would benefit from directly accessing the data but that do not have the technical skill or knowledge required.

Customer data mapping documentation can be used on data profiling, harmonization, enrichment, monitoring, cleansing and in migration for identifying which database entities should be included.

9 Case "Seventh Heaven"

This chapter aims at giving a practical example of how to create and maintain the data mapping between the data model and data storage.

An imaginary company named Seventh Heaven is presented in this thesis. This company is working in the field of Cable TV offering services to its customers, as their services offer different packages of channels and related electronic equipment. The cable network is rented from another company.

To be able to run business, Seventh Heaven has:

- A web store used by customers to order channels and record customer information
- A CRM application to maintain customer information which is also used as the second ordering channel
- A billing system for holding product prices, invoicing and accounting
- A provisioning system, for configuring customers' access to their purchased channels in the network
- ERP for delivering equipment to the customer and for managing equipment stock
- An accounting system for its own financial data
- An HR system with employee records

These are supported by a network infrastructure but its details will be left out as they are not relevant for this example. They have taken ITIL processes to give support to the work.

All starts from the service strategy process. The goal is to reduce incorrect data in the systems to be able to send equipment to correct customers, send invoices to correct addresses and reduce financial loss caused by the previously mentioned issues. These corrections have the added benefit of higher customer satisfaction and less work load on system users. Correct and synchronized data will also support the planned future development of an UI providing a single view of customer data (now customer data can be viewed as separate pieces in the various systems). The MDM department is

discussing the company strategy of improving its monitoring systems with business side and wants to implement data quality measurement tools with performance indicators.

The business department has defined the conceptual model and in cooperation with the IT department they have created a logical model of a customer which is presented in figure 9.

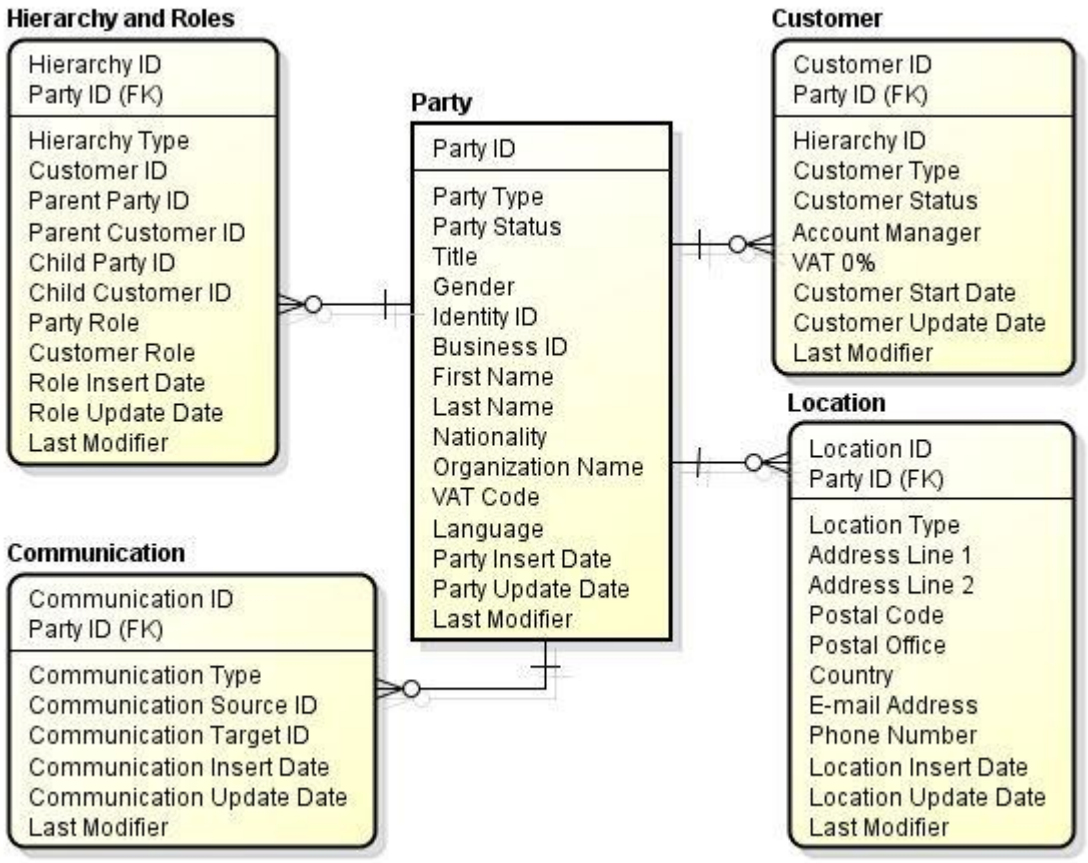


Figure 9. "Seveth Heaven" logical model of the customer

In figure 9, party is the main entity. For example, the party might have many addresses which are stored in the location entity.

9.1 Responsibilities in the Process

One of the ITIL main principles is responsibility. This means that dedicated responsibility is appointed to take required actions and to ensure that handover is done from one process to another.

The **data model owner** takes care of the conceptual model and the data model. She/he communicates with different user groups about requirements of the model. Her/his role is to develop the model as a mirror of business needs. She/he is working close to users of the systems.

Demand management has the role to check that requests are completed, understand the business need and discuss with capacity management that needed resources, such as financial and human resources, are available to meet the request.

The system manager takes care of the vendor management. She/he has deep understanding of system functionalities.

The MDM department has to assure data quality and their interest is to define, with the data model responsible, the data mapping service request to co-operate with system managers and vendors to implement the data mapping service request and implement the continual service improvement phase. The service improvement phase means improving their data measurement tools to get more information about data issues and to use this information for creating and implementing new service requests.

Software vendors have developers which are building and developing their systems. They have appointed requests to change functionalities of their systems. The vendor's general role is to fulfill the service request, implement it in production and take care of it in service operation.

Analysts produce various kinds of reports for sales, accounting or marketing, based on data fetched from IT systems.

Service Desk is the first contact point in case of incidents in the system. They handle incident management and problem management. For them it is essential to know what kind of features the system has.

Change Advisory Board (CAB) consists of change managers who have the responsibility of risk assessment in the business processes. They consider the effect of changes upon the infrastructure, customer service and other services as well.

9.2 Starting Point

The process of creating data model mapping starts from the Business Service Strategy. A customer data model responsible is nominated on the business side. Process flow is described in figure 10 and it is also included in appendix 3 for closer examination.

The data mapping process starts from the service strategy's targets of getting harmonized data, more efficient use of data and getting data profiled. In case of migration or consolidation of systems, the quality of the data elements will be known and the areas of improvement will be identified.

Based on the needs identified in the service strategy, the customer data model mapping is requested by the customer model responsible.

At first, the request is evaluated by the demand manager who will check if the request is complete. If it is complete, then the cost of the request implementation is taken into account in the budget and if the staff capable of fulfilling the task is available the request will be scheduled. In this case, the demand manager asks from system managers and the MDM department if they have time to fulfill the request. At the same time, the demand manager asks from system managers for financial facts, if the budget to invest in this activity on the vendor side is available. If all these assets are met, then the request is passed to the MDM manager.

The main interest in implementing the mapping is on the MDM side. At the beginning, MDM makes a project plan, defines timelines and workload required to complete the mapping. Based on this information, system managers will order needed resources from vendors.

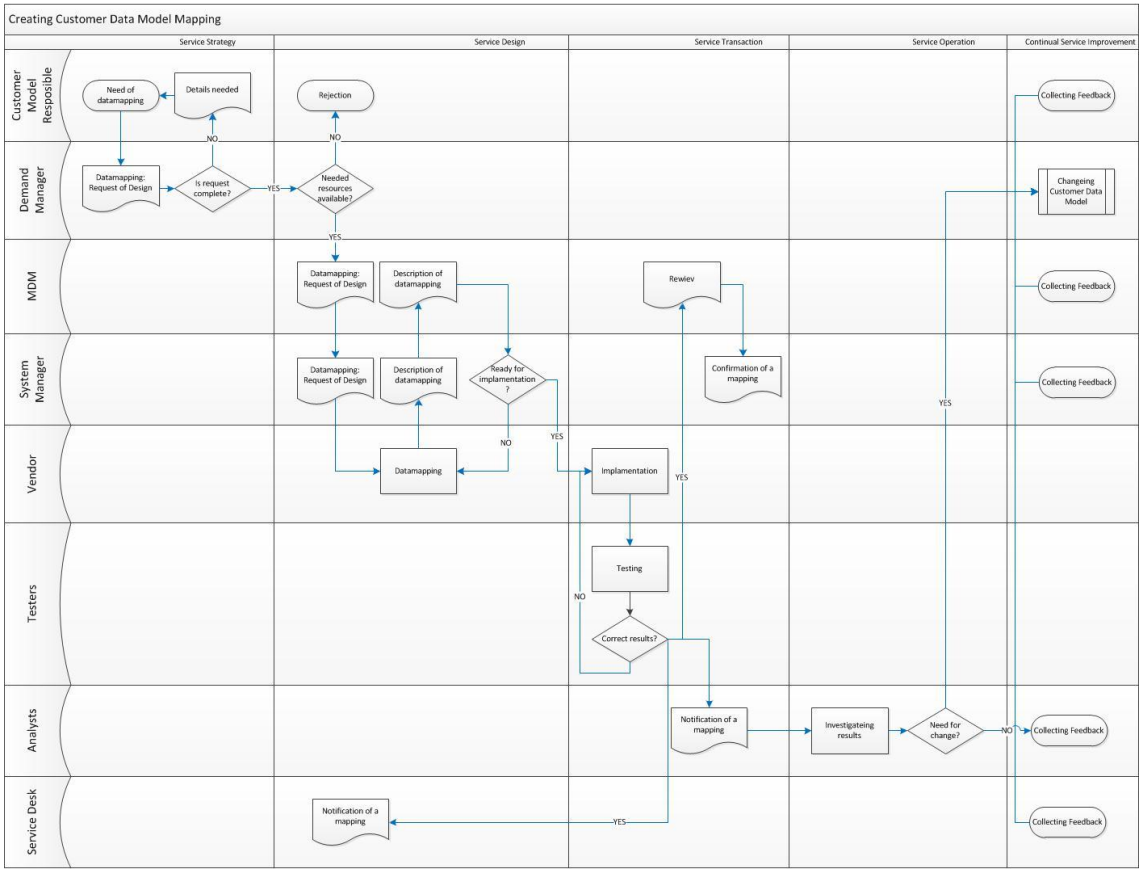


Figure 10. Process of creating and maintaining customer data model

At this stage the key persons for the kick off meeting should be identified. People who can decide resource allocation on the vendor side have a good understanding of the department employees and their capacity of working with issue. The following need to be prepared for the meeting:

- A presentation of the project containing a short history of the project and what the project aims are
- The workspace (tools for recording tasks, setting meetings, document repository and other such tools)
- Prerequisite material in the workspace (such as technical documentation of the systems holding data)
- A folder for moments of the meeting, MoMs can be stored here to reduce communication overhead
- The presentation material and a location for keeping work results (for now the mapping template)

9.3 First Meeting

The purpose of the first meeting is to present the mapping activity, why the logical customer model must be implemented, and what the aim of the project is and to commit the department to work on the project. In the process (see figure 10), this meeting takes place in the service design flow when the data mapping request for design is passed from the system manager to the vendor.

The benefits of the activity, such as assisting other teams, reducing future workload, more transparency, easier future system development, and simpler processes, must be presented to convince the team of the value of this task.
Except for this, the following points must be clarified:

- Agree on the timetable, what should be done and when, such as handling one logical model entity per week.
- Identify the key persons who will be working on particular areas. Vendor managers have the knowledge of whom to appoint to the task.
- Agree on the work methods and check that the people have the needed tools for their work, such as correct workspace.
- Hand over prerequisite material or workspace locations where this was stored
- Point out any logical model related questions to the business team contact people.
- Keep up minutes of the meeting including:

    - date and time
    - participants
    - what is agreed on, who is liable for agreed points
    - date of next meeting, place and subject of the meeting
    - distribution list

After the meeting, access to the workspace must be given to the people who are agreed to be working with the data mapping. It is a good idea to send requests to follow up meetings when action points are still fresh in memory. These meetings ensure commitment to the work timetable.

9.4 Follow Up Meetings

Follow up meetings must be scheduled based on the agreed project plan. They should include all people who are working on the task, which includes system developers and database architects, to discuss the results of mapping with the business team.

- Review mapping results of the agreed entity
- Discuss the results of the work done according appendix 1, if not clear – make appointments for workshops where these can be tested and discussed in detail
- Keep minutes of the meeting

The testing of the SQL queries part of the mapping can be done after the meetings where the data mapping results were reviewed. This part of the process can be seen in figure 9 in the service design phase where the task is discussed with the vendor, system manager and MDM. It can be agreed that each meeting has a theme of one entity so that the handling of the entity is easier to hand over from the service design process to the service transaction process, where the data mapping is implemented.

9.5 Mapping Examples between Logical Model and Database

Systems have different database and table structures. While doing the mapping, the differences became visible.

Table 2. Last Name mapping example between entity Customer, CRM and Billing

Logical Level		
Entity	Customer	
Attribute name	Last name	
Primary Key	Party ID	
Type	Varchar2	
Length	40	
Description	Persons last name	
Physical Level		
Database	CRM	Billing
Table	Contact	Customer
Field	LastName	LastName
Type	Varchar2	Varchar2
Length	40	40
Format	-	-
Description	-	-
Mandatory	Yes	Yes
SQL	SELECT LastName FROM Contact;	SELECT LastName FROM Customer;
Example	SELECT LastName FROM Contact WHERE ID = 123; Result: Kokkonen	SELECT LastName FROM Customer WHERE ID = 123; Result: Kokkonen
UI Field	Last name	-
Picture of UI Field	-	-
Notes	-	-

Some of the attributes were found directly as can be seen in table 2 which contains information about the entity Party and the attribute Last Name.

In table 3 it can be seen that the CRM database had an attribute, Address Line 1 of entity Location, as it was described in Customer Logical Model, but on the billing side Address Line 1 was divided into individual columns on the Address table. Those independent data items on the billing side had to be concatenated to retrieve the wanted result.

Table 3. Address mapping example between entity Location and CRM and Billing

Logical Level		
Entity	Location	
Attribute name	Address Line1	
Primary Key	Location ID	
Type	Varchar2	
Length	40	
Description	Primary address	
Physical Level		
Database	CRM	Billing
Table	Address	Address
Field	AddressLine1	Street, StreetNro, Starecase, Flat
Type	Varchar2	Varchar2
Length	40	40,10,10,10
Format	-	-
Description	-	-
Mandatory	Yes	Street Yes
SQL	SELECT AddressLine1 FROM Address;	SELECT Street || ' ' || StreetNro || ' ' || Starecase || ' ' || Flat FROM Address;
Example	SELECT AddressLine1 FROM AddressWHERE ID = 123; Result: Kauppakatu 5 A 7	SELECT Street || ' ' || StreetNro || ' ' || Starecase || ' ' || Flat FROM Address WHERE AddressID = 123; Result: Kauppakatu 5 A 7
UI Field	Street address	-
Picture of UI Field	-	-
Notes	-	-

The dataflow is from the CRM to the billing system. In this case there must be some internal logic of splitting data as Address Line 1 can be seen divided into billing side table with several columns.

The entity Location attribute Phone number was placed in the CRM database in the Address table while the billing database had a different logic and had it placed as a part of the Customer table. This can be seen in table 4.

Table 4. Phone Number mapping example between entity Location, CRM and Billing

Logical Level		
Entity	Location	
Attribute name	Phone number	
Primary Key	Location ID	
Type	Varchar2	
Length	20	
Description	Phone number, land line or mobile number	
Physical Level		
Database	CRM	Billing
Table	Address	Customer
Field	PhoneNumber	Phone
Type	Varchar2	Number
Length	15	15
Format	-	-
Description	-	-
Mandatory	No	No
SQL	SELECT PhoneNumber FROM Addess;	SELECT Phone FROM Customer;
Example	SELECT PhoneNumber FROM Address WHERE ID = 123; Result: +358501231234	SELECT Phone FROM Customer WHERE ID= 123; Result: 358501231234
UI Field	Contact number	-
Picture of UI Field	-	-
Notes	-	-

During the dataflow from the CRM database to billing, the phone number is changed. On the CRM side, the type of the phone number is Varchar2, and it allows number +358501231234, to be inserted. However on the billing side, it allows just numbers so modification is needed before inserting a phone number in the billing DB.

The CRM and billing systems have been designed for different purposes. More differences are found as observed in table 5.

Table 5. Communication Type mapping example between entity Communication and CRM and Billing

Logical Level		
Entity	Communication	
Attribute name	Communication Type	
Primary Key	Communication ID	
Type	Varchar2	
Length	40	
Description	Type of communication; Letter, Invoice, Call, E-mail, SMS	
Physical Level		
Database	CRM	Billing
Table	Interaction	-
Field	Type	-
Type	Varchar2	-
Length	10	-
Format	-	-
Description	-	-
Mandatory	No	-
SQL	SELECT Type FROM Interaction;	-
Example	SELECT DISTINCT Type FROM Interaction; Result: Letter, Call, Invoice	-
UI Field	Interaction type	-
Picture of UI Field	-	-
Notes	-	-

It can be seen in table 5 that the attribute Communication Type of the entity Communication is missing from the billing database, but, as one of the functions of CRM is to keep record of interactions with customers, the attribute was found.

9.6 Final Review

At this point all data is gathered and documented by agreed persons. In this case, the MDM department took the responsibility to create a master document and collect all needed information from the workspace. At the beginning of the produced document there is a brief introduction to the business strategy and customer logical model followed by a short introduction to the entities of the system. After these preconditions are presented, the mapping between the logical model and various systems is added. This work is presented to the target groups who were working with the mapping or people that might use it. From the process point of view this is the service transaction process.

- Review the results of the work as documented in appendxix 1 – is the mapping information verified and approved?
- Agree on how the mapping should be published, including place and time
- Agree on who will present the mapping to employees, such as database architects, system developers, who will be maintaining it
- Agree on who will present the mapping to employees and who will be using it as a tool, for example analysts, service desk
- Create a MoM

9.7 Handover to Production Phase

Once the mapping is done and the document created, its lifecycle moves from the Service Transaction process to Service Operation. This is the process step Notification of a Mapping from the Service Transaction flow (see figure 9) and it is about committing employees to work based on the agreed Customer Data Model Mapping. At this phase of the process more people are involved, not only the development side but also the service operation such as service desk which are monitoring the behavior of systems based on incidents and analysts to improve their reporting.

While working with the document, developers and architects got a picture of what is expected from them in the future, such as checking if the code is violating the Customer Data Model, or if the Customer Data Model changes, how it will impact DB

schemas, how it should be changed, and whom to contact in case of issues. However analysts and service desk operators might not have that information. The mapping must be presented to the service desk employees together with the benefits they can gain by using it, such as being able to check that the field value and formatting are correct and in case of errors being able to provide a better root cause analysis. Analysts can use it to achieve better understanding of the data locations, enabling them to provide more complex reports.

This meeting follows the same general guidelines as the first meeting: present the idea, and commit to the idea and appoint the responsibilities to the employees.A MoM will be created and DMS access given to the people who will be working with the mapping.

9.8 Maintaining and Changing a Data Mapping

Isaac Asimov said:

> It is change, continuing change, inevitable change, that is the dominant factor in society today. No sensible decision can be made any longer without taking into account not only the world as it is, but the world as it will be...
> [13]

This quote is particularly true in the business environment of today. Business needs change and the development of systems with it. After implementing the customer data model mapping as a tool for users, it needs to be updated with all changes which are done to the data structure of the production environment. These updates have to be done in the system development phases, when code affecting application data is changed (such as SQL queries).

The need for a change might come during the Service Operation process when a system behavior flaw is found. For example, a new customer named "Me & Co." is inserted by a self-service application user and from there the data is transmitted using the XML encoding (which is designed to transport, share and display information between applications or via the application to the database) to CRM. At the CRM side it is stored as "Me &amp Co." instead of Me & Co, because XML has special use for some predefined characters (&, ", ', <,>). The customer would not be delighted to get a bill on the name Me &amp Co. Similar issues can be caused by other interface mismatches, for example ISO-8859-1 does not support the euro sign (€).This means

that the different interfaces and their data format need to be considered when developing systems.

Creating and changing the data mapping includes differences. From needs of various groups improvement change requests are done on continual service improvement process. When figure 10 is compared to figure 11, one additional team called Change Advisory Board (CAB) is mentioned. The CAB responsible is to make sure that any impacts of the changes on the system or customer data model are taken into account on the other systems which have a relation to them. Figure 10 and figure 11 are also included as attachments for closer examination. In the earlier process, implementing the customer data mapping to existing systems has itself no impact as it is just about finding relations between the logical model and the system, not making any change in them. However after analyzing the data of the mapping results, improvements are made which might cause change requests for the mapping and analyzed mapping results might also include changing systems or its database. The business strategy is guiding the business requirements. The creation of Customer Data Model Mapping is a top-down process but maintaining it can be initiated from any point (logical model changes, data storage changes, or application changes).

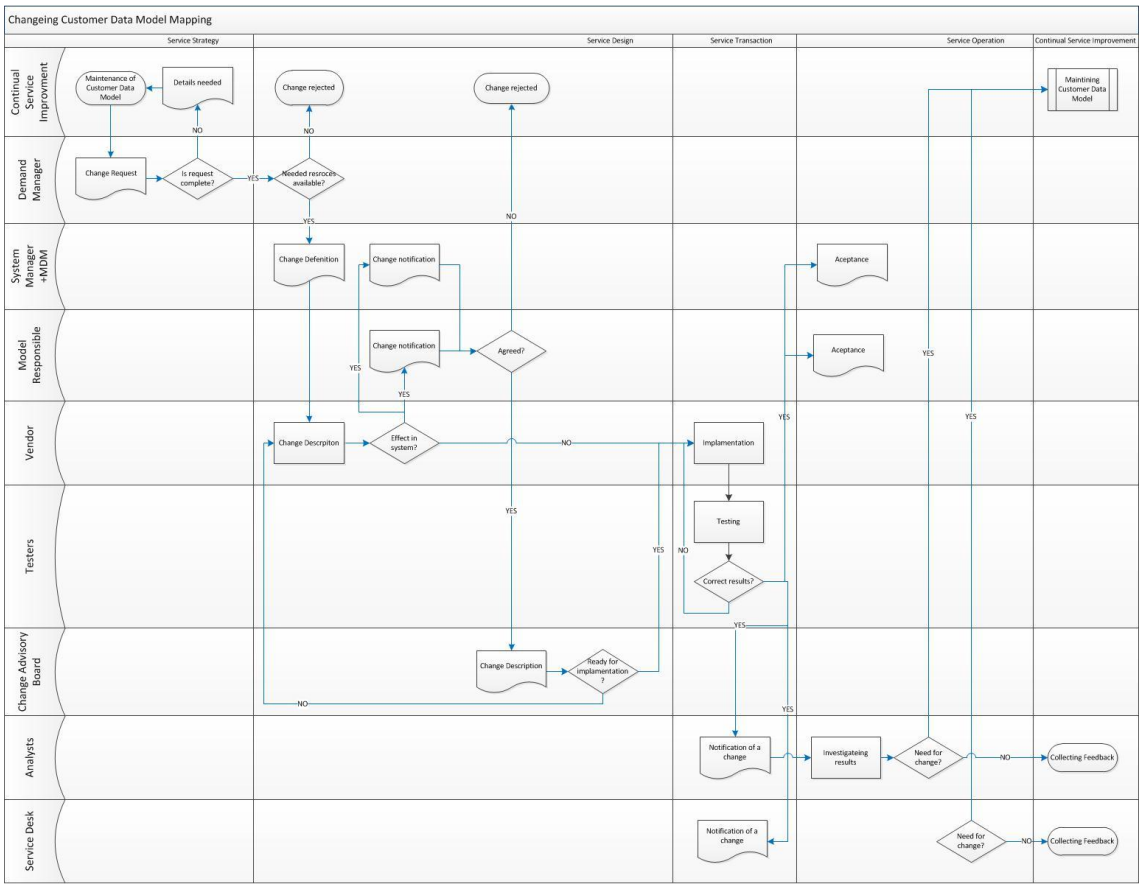


Figure 11. Change flow of the customer model mapping

For changes of the logical model, the business teams have a bigger role in the activity. If some of the customer logical data model entity attributes are changed and it has an impact on database values, a detailed analysis must be done for the cleanup and enrichment of affected data.

9.9 Challenges Encountered

If the system vendor has full responsibility of the applications and databases, communication gaps and language barriers might lead to suspicion and much focus around questions like "Why we are doing this?" "What is the purpose of this?". Care must be taken to assure that the vendor fully co-operates and, since this kind of task is done on top of regular system development or maintenance work, it might take more time than expected for the whole team needed to get together and go through all the entity attributes. If the system is maintained by the organization directly, the commitment and time issues can be solved more easily.

It can take a while to reach a common understanding of the meaning of the entity attributes in the logical model. The Customer Data Model is used infrequently, and even if a document is available, some fields will probably raise a lot of discussion. A series of internal meetings might be needed to make attribute definitions clear before having the meetings with the vendor.

Verification of the information provided by the vendor is usually quite fluent since direct SQL queries can be used on the database. However, some of the attributes can be difficult to test if common understanding in the mapping team is lacking of the meaning of that entity attribute. These issues will be accentuated in cases where the logical model and applications do not contain the same logical entities. For example, the Party entity of the logical model can be a consumer or a corporate customer which does not necessarily have an agreement. The system has two related entities: contact and account. In an account there are contacts with different roles and usually a contact is linked to an account, but this is not mandatory. On the other hand, an account cannot exist without a contact linked to it but an account can exist without an agreement attached to it. The vendor might consider that the account is a party, not the contact itself which business considers being a party. With co-operation these issues can be worked out.

Another challenge comes from dealing with production database environment for information verification. If the person performing the verification comes from the business side it might be difficult to deal with the various SQL queries that need to be used. Also, the vendor database documentation might be either too technical, closer to the internal workings of the application rather than to the application's functionality, not updated or missing altogether. The security and performance restriction of the environment might delay this verification too.

10 Conclusion

One of my work assignments was to make a clear documentation for the mapping between the Customer Logical Model and the data in the CRM and Billing systems. In addition process to create and maintain this mapping was needed. Generally speaking most of the IT work is outsourced in companies running IT business systems and the teams doing the monitoring and related processes do not have deep knowledge of the systems they are working with.

During the work assignment I followed the same steps as defined in this document, eventually having to study if the Entity attributes were available in the systems and in cases where they were missing if the information could be accessed from third party systems. The result was surprising: some of the attributes of the entities described in the work assignment customer model were not used at all even though they were defined as business critical information.

Some of the logical model attributes can be left out altogether, such as attribute longitude and latitude in the location entity. This is relevant information when we have a property outside the infrastructure. If locations with geographical coordinates must be recorded in the system, then a generic reference field can be used to store this additional information. If these attributes are indeed needed, there should be an automated way of populating them, such as an address conversion to geographic coordinates, as it is very unlikely that all customers will provide geographic coordinates when beginning contact with the organization.

Some data was retrieved from third party systems but it was not updated in the studied IT systems. This raises the question of the image one has of their customers. Are several applications needed to understand a customer well? For customer interaction it is important that the user gets access to customer data fast and easily, from a single source. This highlights the importance of MDM, of creating different kinds of cross-system views and of enabling system development according to the Customer data model.

Resources are often wasted in the service operation process if insufficient effort is put into design at the beginning of service development. Being agile does not mean that the importance of documentation can be ignored. Unfortunately, immediate monetary gains are usually the main driver setting the rules of production. The value to business is to develop new services for getting more ROI (Return of Investment) which pushes the development side to deliver more at the cost of quality.

Cleaning data is not seen as an important activity and not enough resources are invested in it. The "extinguishing fire" method is more commonly used. If some problem is noted and Root Cause Analysis (RCA) is made which is pointing to invalid data, then that particular data issue is rectified. Everyone is waiting for bigger projects as migrations of systems to take care of the data cleansing. If the quality of data is taken seriously the need for doing data correction and the costs of ownership would decrease and customers and employees would both be more satisfied. This can be achieved by regularly measuring data using KPIs and variance is detected, corrective action is taken.

Working on this also demonstrated how dynamic the working environment is. Techniques and methods for solving issues are developed all the time but assimilating and using them is a slow process. In addition, changes in the company strategy or organization require handing over of tasks, which is a slow process.

Data mapping is done in the continual service improvement phase. This means that no changes were done on the database side or in the customer data model during the process. It was done by "walking through the practices", such as observing, gathering data, testing, writing and interviewing employees working with data and business.

Defining a process to keep up the data mapping documents is the responsibility of system developers and system managers. They are the persons with the best knowledge of the system and if there are some new features which have an impact on existing database definitions, only they have the information of both the change and its impact.

Knowledge of Customer Data Model should be increased throughout the company by providing information about it proactively to key persons in various teams.

All of the company's technical documents should be kept in the same repository. In this way it is easy to find related documents as well as agreements defining Service Level Agreements (SLA) or other asset management documents.

Mapping was done on spreadsheets as the initial tool and could be easily implemented in a dedicated mapping tool such as ErWin Datamodeller, SAP Sybase PowerDesigner or the open source system Open ModelSphere. These allow generating an MDM as part of the continual process. These tools will surely be used in the future and then the quality of the data will increase.

References

1 International Organization for Standardization (ISO). About ISO [online]. URL: http://www.iso.org/iso/home/about.htm. Accessed 11 March 2013.

2 Loshin, David. Master data management. New York: Morgan Kaufmann; 2008.

3 Knigh, Brian, Veerman, Erik, Moss, Jessica M, Davis, Mike and Rock, Chris. Professional Microsoft SQL Server 2012 integration services. Wrox; 2012.

4 Loshin, David. The practitioner's guide to data quality improvement. New York; Morgan Kaufmann; 2010.

5 Cervo, Dalton, Allen, Mark. Master data management in practice. New Jersey: John Wiley & Sons Pub; 2011.

6 Hoberman, Steve, Burbank, Donna, Bradley, Chris. Data modeling for the business: A handbook for aligning the business with IT using high-level data models. Bradley Beach: Technics Publications; 2009.

7 Oppel ,Andy. Data modeling. USA: McGraw-Hill; 2009.

8 English, Bill, Alderman, Brian, Ferraz, Mark. Microsoft® SharePoint® 2010 Administrator's Companion. Redmond, Washington: Microsoft Press; 2010.

9 Ganci, John, Fox, Leonard, Hutton, James, Jensen, Thomas, Stec, Andrzej, Zhang, Eric. Document management using WebSphere Portal V5.0.2 and DB2 Content Manager V8. IBM Redbooks; 2004.

10 The official introduction to the ITIL service lifecycle. Norwich: The Stationery Office; 2007.

11 Kriegel, Alex, Trukhnov, Boris M. SQL Bible, Second Edition. Indianapolis: Wiley Publishing Inc; 2008.

12 Singh, Shio Kumar. Database systems: Concepts, design and applications, Second Edition. New Delhi: Dorling Kinderley; 2011.

13 Wikipedia. Isaac Asimov [online]. URL: http://en.wikiquote.org/wiki/Isaac_Asimov. Accessed 11 March 2013.

14 Kriegel, Alex. Discovering SQL. Indianapolis: Wiley Publishing Inc; 2011.

15 Choate, Mark. Professional wikis. Indeanapolis: Wiley Publishing Inc; 2008.

Seventh Heaven Logical Customer Model

	UI		Pysical Model										Business Definition	Logical Model				Conceptual Model
Notes	Picture of the UI Field	UI Field	Example	SQL Quer	Mandotory	Description	Format	Length	Type	Field	Table	System	Description	Primary Key /	Length	Type	Attribute	**Entity**
																		Party
													System provided unique identifier of a person or a company	PK	10	Number	Party ID	
													Identifies if party is a corporate or a person		10	Varchar2	Party Type	
													Active/Inactive		10	Varchar2	Party Status	
													Miss/Mr/Mrs		10	Varchar2	Title	
													Male/Female		10	Varchar2	Gender	
													Unique personal identity code given by state authority		10	Varchar2	Identity ID	
													Unique national identity code for companies given by state authority		10	Varchar3	Business ID	
													Last name of the person		40	Varchar2	Last Name	
													First name of the person		20	Varchar2	First Name	
													Nationality		15	Varchar2	Nationality	
													Name of the Corporate		30	Varchar2	Organization Name	
													The Value added tax identification number is used European Union Value Added Tax for taxation		10	Varchar2	VAT Code	
													Party communication language		20	Varchar2	Language	
													System provided date when information is inserted			Date	Party Insert Date	
													System provided date when information is changed			Date	Party Update	
													Identifier person or a system who commited updated		10	Varchar2	Last Modifier	

	UI		Pysical Model										Business Definition	Logical Model				Conceptual Model
Notes	Picture of the UI Field	UI Field	Exam ple	SQL Quer	Mand otory	Descri ption	Format	Length	Type	Field	Table	Syst em	Description	Primary Key /	Length	Type	Attribute	**Entity**
																		Customer
													System provided unique identifier of a person or a company when first agreement is made	PK	10	Number	Customer ID	
													System provided unique identifier of a person or a company	FK1	10	Varchar2	Party ID	
													System provided unique identifier of a hierarchy		10	Number	Hierarchy ID	
													Business predefined type of the customer; Residental, Corporate,Entrepreneur		10	Varchar2	Customer Type	
													Active/Inactive		10	Varchar2	Customer Status	
													Person who is responsible of the customer		10	Varchar2	Account Manager	
													Tax exemption		10	Varchar2	VAT 0%	
													System provided date when information is inserted			Date	Customer Start Date	
													System provided date when information is changed			Date	Customer Update	
													Identifier person or a system who commited updated		10	Varchar2	Last Modifier	

	UI		Pysical Model										Business Definition	Logical Model				Conceptual Model
Notes	Picture of the UI Field	UI Field	Exam ple	SQL Quer	Mand otory	Descri ption	Format	Length	Type	Field	Table	Syst em	Description	Primary Key /	Length	Type	Attribute	**Entity**
																		Hierarchy and Roles
													System provided unique identifier of a hierarchy	PK		Number	Hierarchy ID	
													System provided unique identifier of a person or a company	FK1	10	Varchar2	Party ID	
													Indentifier nature of the hierarchy; Legal structure, organization hierarchy		10	Varchar2	Hierarchy Type	
													System provided unique identifier of a person or a company			Number	Parent Party ID	
													System provided unique identifier of a person or a company when first agreement is made			Number	Parent Customer ID	
													System provided unique identifier of a subhierachy of a party			Number	Child Party ID	
													System provided unique identifier of a subhierachy of a customer			Number	Child Customer ID	
													Indentifies the role of the Party; User, Owner, Subsidiary		10	Varchar2	Party Role	
																	Customer Role	
													System provided date when information is inserted			Date	Role Insert Date	
													System provided date when information is changed			Date	Role Update Date	
													Identifier person or a system who commited updated		10	Varchar2	Last Modifier	

	UI		Pysical Model										Business Definition	Logical Model				Conceptual Model
Notes	Picture of the UI Field	UI Field	Example	SQL Quer	Mandotory	Description	Format	Length	Type	Field	Table	System	Description	Primary Key /	Length	Type	Attribute	**Entity**
																		Location
													System provided unique identifier of a location	PK		Number	Location ID	
													System provided unique identifier of a person or a company	FK1		Number	Party ID	
													Shipping / Invoicing location		40	Varchar2	Location Type	
													Primary addressline		40	Varchar2	Address Line 1	
													Secondary addressline		40	Varchar2	Address Line 2	
													Postal Code		10	Varchar2	Postal Code	
													Postal Office Name / City		30	Varchar2	Postal Office	
													Country		20	Varchar2	Country	
													E-mail address		40	Varchar2	E-mail Address	
													Phone number, land line or mobilenumber		20	Varchar2	Phone number	
													System provided date when information is inserted			Date	Location Insert Date	
													System provided date when information is changed			Date	Location Update	
													Identifier person or a system who commited updated		10	Varchar2	Last Modifier	

	UI		Pysical Model										Business Definition	Logical Model				Conceptual Model
																		Communication
													System provided unique identifier of a communication	PK		Number	Communication ID	
													System provided unique identifier of a person or a company	FK1		Number	Party ID	
													Type of communication ; Letter, Invoice, Call, E-mail,SMS		40	Varchar2	Communication Type	
													Defines party ID contacting		10	Varchar2	Communication Source	
													Defines party ID contacted		10	Varchar2	Communication Target	
													System provided date when information is inserted			Date	Communication Insert	
													System provided date when information is changed			Date	Communication Update	
													Identifier person or a system who commited updated		10	Varchar2	Last Modifier	

Seventh Heaven Database Tables

CRM		

Contact		
ID	Number	10
Type	Varchar2	10
Title	Varchar2	10
Gender	Varchar2	10
LastName	Varchar2	40
FirstName	Varchar2	40
Company	Varchar2	40
Nationality	Varchar2	40
SSID	Varchar2	10
BusinessID	Varchar2	10
VATCode	Varchar2	10
Language	Varchar2	10
InsertDate	Date	
UpdateDate	Date	
Modifier	Date	20

Customer		
ID	Number	10
ContactID	Number	10
HierarchyID	Number	10
Type	Varchar2	10
Status	Varchar2	10
ResposibleManager	Varchar2	40
StartDate	Date	
UpdateDate	Date	
Modifier	Number	10

Billing		

Customer		
ID	Number	10
Type	Varchar2	10
Title	Varchar2	10
LastName	Varchar2	40
FirstName	Varchar2	40
OrganizationName	Varchar2	40
SID	Varchar3	10
Status	Varchar2	40
VAT	Varchar2	10
Language	Varchar2	10
Email	Varchar2	40
Phone	Number	15
InsertDate	Date	
UpdateDate	Date	
Modifier	Date	20

Account		
AccuntID	Number	10
CustomerID	Number	10
HierarchyID	Number	10
Type	Varchar2	10
Status	Varchar2	10
VAT0	Varchar2	10
StartDate	Date	
UpdateDate	Date	
Last Modifier	Number	10

CRM		
Address		
ID	Number	10
CustomerID	Number	10
LocationType	Varchar2	10
AddresLine1	Varchar2	40
AddressLine2	Varchar2	40
PostalCode	Varchar2	10
PostalOffice	Varchar2	40
Country	Varchar2	20
EmailAddress	Varchar2	40
Phonenumber	Varchar2	15
InsertDate	Date	
UpdateDate	Date	
Modifier	Varchar2	10

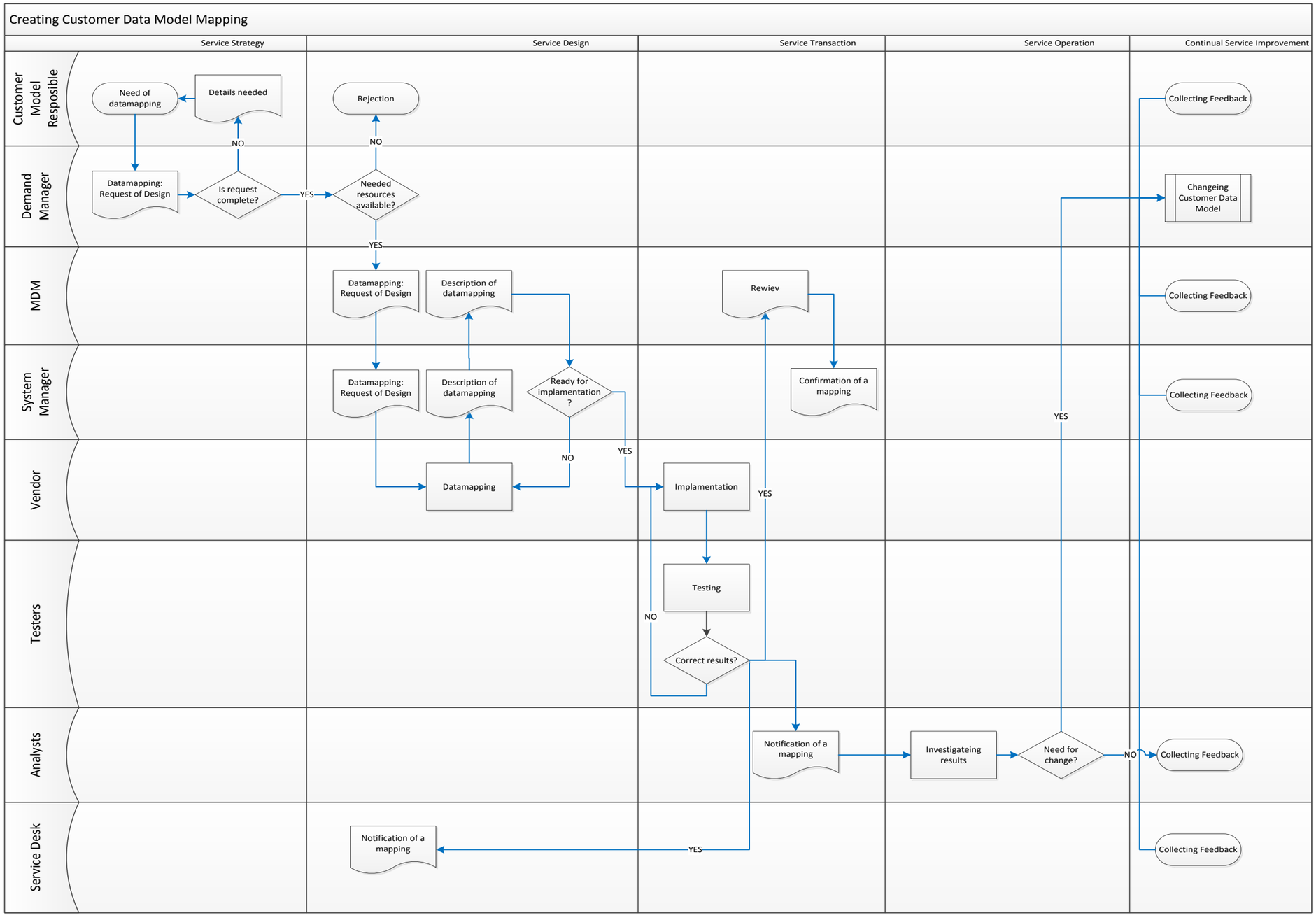
Billing		
Address		
AddressID	Number	10
CustomerID	Number	10
Type	Varchar2	10
POBox	Varchar2	40
Street	Varchar2	40
StreetNro	Varchar2	10
Starecase	Varchar2	10
Flat	Number	10
PostalCode	Varchar2	10
PostalOffice	Varchar2	40
Country	Varchar2	20
InsertDate	Date	
UpdateDate	Date	
Modifier	Varchar2	10

Hierarchy		
ID	Number	10
Type	Varchar2	10
ParentContactID	Number	10
ParentCustomerID	Number	10
ChildContactID	Number	10
ChildCustomerID	Number	10
Role	Varchar2	10
InsertDate	Date	
UpdateDate	Date	
Modifier	Number	10

Hierarchy		
ID	Number	10
Type	Varchar2	10
ParentCustomerID	Number	10
ParentCustomerID	Number	10
ChildAccountID	Number	10
ChildAccountID	Number	10
Role	Varchar2	10
InsertDate	Date	
UpdateDate	Date	
Modifier	Number	10

CRM		
Interaction		
CommunicationID		10
CustomerID		10
Type	Varchar2	10
Source	Number	10
Target	Number	10
InsertDat	Date	
UpdateDa	Date	
Modifier	Varchar2	10

Process of Creating and Maintaining Customer Data Model



Change Flow of the Customer Model Mapping